Training MLPs Layer-by-Layer with the Information Potential

Dongxin Xu, Jose C. Principe

Computational NeuroEngineering Laboratory
NEB 486, Electrical and Computer Engineering Department
University of Florida, Gainesville, FL 32611, USA
(xu, principe)@cnel.ufl.edu

ABSTRACT

In the area of information processing one fundamental issue is how to measure the relationship between two variables based only on their samples. In a previous paper, the idea of ***Information Potential*** which was formulated from the so called ***Quadratic Mutual Information*** was introduced, and successfully applied to problems such as Blind Source Separation and Pose Estimation of SAR (Synthetic Aperture Radar) Images. This paper shows how information potential can be used to train a MLP (multilayer perceptron) layer-by-layer, which provides evidence that the hidden layer of a MLP serves as an “information filter” which tries to best represent the desired output in that layer in the statistical sense of mutual information.

1. INTRODUCTION

Many signal processing problems can be regarded as manipulation of information at the output space of a mapper or its internal hidden space such as the hidden layer of a MLP. In many real world problems, however, the only available information about the domain is contained in the data collected. It is therefore practically significant to estimate the information entropy of a variable or the statistical relation between variables based merely on the given data samples, without further assumption. This paper and the previous ones [1], [2], show how information measures can be related to a “potential energy” of the collected data when the samples are interpreted as physical particles. In fact, data samples are atoms conveying information. For both reasons, a data sample will be called “information particle” (IPT). The potential energy of IPTs is therefore called “information potential” (IP). The derivative of IP with respect to the position of IPT can be interpreted as interactions among IPTs and is thus called “information force” (IF). Variations of information entropy or mutual information are equivalent to the variations of their corresponding IP, which can be implemented by moving IPTs along the direction of the IFs they receive. If one seeks to manipulate information at a certain layer of a network, the IFs can not move IPTs directly but instead can be propagated through the dual network to adapt each parameter.

Information potential is a rather general and non-parametric way to estimate information from data samples. It has wide applications and was successfully applied to Blind Source Separation, Pose Estimation of SAR Images, etc., [1], [2]. We will restrict in this paper our discussion to the problem of MLP training in a layer-by-layer manner.

The much higher computational power of the MLP when compared with the perceptron was recognized many years ago [3]. We know today that the MLP is an universal mapper [4]. However, because of the lack of an efficient training algorithm comparable to the perceptron learning rule, we had to wait until 1985 for the back-propagation (BP) algorithm to harness the power of the MLP. In the BP procedure the output error is first computed and then back-propagated though the dual network to parameters, which meets the goal but raises problems such as lack of biological evidence [5], difficulty in explaining what is really learned in the hidden layer, slow convergence speed, etc.. If the MLP can be trained layer-by-layer, such problems will be alleviated.

Information Theoretic Learning can help us implement MLP training layer-by-layer. Linsker [6] pointed out that a linear or nonlinear network can be treated as an information channel. A principle for self-organizing such networks is to transfer as much information as possible through the network. Linsker’s idea can be extended further to include not only the input information but also other sources of information which are usually characterized by desired outputs. So, from this point view, each layer of a MLP can be regarded as an “information filter”, and the training purpose is to maximize the mutual information between the output of each layer and the desired output. The IP method can thus be used when only samples of inputs and desired outputs are provided.

Finally, it should be pointed out that because of the similarity between an information particle and a physical particle, we believe that this method may be suitable for implementation of Quantum Computing which will facilitate the computation of IP and IF.

2. Quadratic Entropy and IP

Let $a_i \in R^m, i = 1, \ldots, N$, be a set of samples from the output $Y \in R^m$ of a mapping $R^n \to R^m$: $Y = q(X, \theta)$, where θ is a set of parameters. Based on Renyi's Quadratic Entropy and Parzen Window pdf estimation with Gaussian Kernel, the idea of IP and formula (1) can be derived.

$$\begin{cases} R_2(Y|a) = -\log\int f_Y(y)^2 dy = -\log V(a) \\ V(a) = \dfrac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} G(a_i - a_j, 2\sigma^2) \end{cases} \tag{1}$$

where $G(y, \sigma^2) = \exp(-(y^T y)/(2\sigma^2))/((2\pi)^{m/2}\sigma^m)$, $f_Y(y) = \sum_{i=1}^{N} G(y - a_i, \sigma^2)/N$ is the Parzen Window estimation of pdf for Y, $R_2(Y|a)$ is the estimation of Renyi's Quadratic Entropy on a data set a. $V(a)$ is called Information Potential (IP) because each term in the summation can be interpreted as an potential between two particles a_i and a_j. As pointed out in [1], Renyi's entropy is equivalent to Shannon's entropy with regards to entropy maximization. So, maximization of entropy is equivalent to minimization of $V(a)$. The derivative of IP with respect to each parameter θ of the mapping can be decomposed into two parts as (2), where $f_i = \partial V/\partial a_i$ is the "information force" (IF) that particle a_i receives, $\partial a_i/\partial\theta$ is the sensitivity of a_i with respect to each parameter θ. Notice that both f_i and $\partial a_i/\partial\theta$ are vectors. By analogy with error back-propagation [7], equation (2) can be interpreted as information force back-propagation (IFBP), where sensitivities $\partial a_i/\partial\theta$ serve as the "transmission mechanism" through which IFs are back-propagated to the parameter θ.

$$\frac{\partial}{\partial\theta}V(a) = \sum_{i=1}^{N}(f_i)^T\frac{\partial a_i}{\partial\theta}, \qquad f_i = \frac{\partial}{\partial a_i}V(a) \tag{2}$$

Once the forces are calculated, they will be back-propagated exactly the same way as error back-propagation. To calculate IP and IF, two matrices can be defined as (3) and their structures are illustrated in Figure 1.

$$\begin{cases} D = \{d_{ij}\}, \qquad d_{ij} = a_i - a_j \\ V = \{v_{ij}\}, \qquad v_{ij} = G(d_{ij}, 2\sigma^2) \end{cases} \tag{3}$$

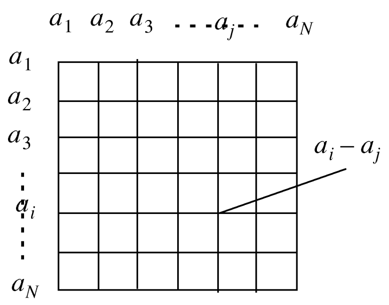

Figure 1. The structure of Matrix D & V

Notice that each element of D is a vector in R^m space while each element of V is a scalar. It is easy to show that

$$\begin{aligned} V(a) &= v = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij} \\ f_i &= \frac{-1}{N^2\sigma^2}\sum_{j=1}^{N} v_{ij}d_{ij} \qquad i = 1, \ldots, N \end{aligned} \tag{4}$$

We can also define the IP for each particle a_i as: $v_i = \frac{1}{N}\sum_{i=1}^{N} v_{ij}$. Obviously, $v = \frac{1}{N}\sum_{i=1}^{N} v_i$

3. Quadratic Mutual Information & CIP

As section 2 indicates, the quadratic form of a pdf can be simplified as an IP when the Parzen Window is used for the estimation of the pdf. Unfortunately, the mutual information based on both Kullback-Leibler divergence and Renyi's divergence is not quadratic on the pdfs [1]. So, based on the Cauchy-Schwartz inequality $\|a\|^2\|b\|^2 \ge (a^T b)^2$ where quadratic forms are basic terms, a measure of statistical relation between two random variables which is called Cauchy-Schwartz Quadratic Mutual Information (CS-QMI) was proposed in [1]. In this paper, a simpler inequality $\|a - b\|^2 = \|a\|^2 + \|b\|^2 - 2a^T b \ge 0$ is used to construct a similar measure $D(Y^1, Y^2) =$

$$\begin{aligned} &\iint f_{Y^1Y^2}(y^1, y^2)^2 dy^1 dy^2 + \iint f_{Y^1}(y^1)^2 f_{Y^2}(y^2)^2 dy^1 dy^2 \\ &- 2\iint f_{Y^1Y^2}(y^1, y^2) f_{Y^1}(y^1) f_{Y^2}(y^2) dy^1 dy^2 \end{aligned} .$$

where Y^1 and Y^2 are two random variables with joint pdf $f_{Y^1Y^2}(y^1, y^2)$ and marginal pdfs $f_{Y^1}(y^1)$ and $f_{Y^2}(y^2)$. Obviously, $D(Y^1, Y^2) \ge 0$ and equality holds if and only if Y^1

and Y^2 are statistically independent. Basically, it measures the Euclidean distance between joint pdf $f_{Y^1Y^2}(y^1, y^2)$ and factorized marginal pdfs $f_{Y^1}(y^1)f_{Y^2}(y^2)$, and thus will be called Euclidean Distance Quadratic Mutual Information (ED-QMI). As [1] has shown, it is evidentially a measure for independence, and [2] also shows experimentally its validity as an dependence measure.

If a set of data samples for joint variable $(Y^1, Y^2)^T$ is: $a = \{a_i = (a_i^1, a_i^2)^T,\ i = 1, \dots, N\}$ The Parzen Window estimation of joint pdf and marginal pdfs will be:

$$
\begin{aligned}
f_{Y^1Y^2}(y^1, y^2) &= \frac{1}{N}\sum_{i=1}^{N} G(y^1 - a_i^1, \sigma^2)G(y^2 - a_i^2, \sigma^2) \\
f_{Y^l}(y^l) &= \frac{1}{N}\sum_{i=1}^{N} G(y^l - a_i^l, \sigma^2)\ , \quad l = 1, 2
\end{aligned}
\tag{5}
$$

With (5), it is not difficult to obtain (6):

$$
\begin{aligned}
D((Y^1, Y^2)|a) &= V_c(a) \\
V_c(a) = v_c &= \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} v_{ij}^1 v_{ij}^2 - \frac{2}{N}\sum_{i=1}^{N} v_i^1 v_i^2 + v^1 v^2
\end{aligned}
\tag{6}
$$

where all $v_{ij}^l,\ v_i^l,\ v^l,\ l = 1, 2$ are similarly obtained as $v_{ij},\ v_i, v$ in section 2 respectively. $l = 1, 2$ is the index for marginal spaces and the names for v_{ij}^l, v_i^l, v^l will be prefixed with "marginal" correspondingly. v_c is called "cross information potential" (CIP). Actually, from (6), we can see that v_c is an overall measure of cross-correlation between two marginal IPs ($\Sigma\Sigma(v_{ij}^1 v_{ij}^2)$, $\Sigma(v_i^1 v_i^2)$ and $v^1 v^2$ are cross-correlations at different levels). So, maximizing ED-QMI $D((Y^1, Y^2)|a)$ is equivalent to maximizing v_c and minimizing $D((Y^1, Y^2)|a)$ is equivalent to minimizing v_c. Since $v_{ij}^k = v_{ji}^k,\ k = 1, 2$, $N\Sigma_{i=1}^N v_i^1 v_i^2$ $= \Sigma_{i=1}^N \Sigma_{j=1}^N v_i^1 v_{ij}^2$ $= \Sigma_{i=1}^N \Sigma_{j=1}^N v_j^1 v_{ij}^2$, etc.. Thus the CIP and the IFs in CIP field can be calculated as (7), where C^k are cross matrices which serve as modifiers and this can be clarified by the similarity between (7) and (4). Equation (7) just states that the CIP can be calculated as an marginal IP modified by the corresponding cross matrix and the marginal IF of the CIP field can be computed by the IF of the original marginal IP field modified by the elements of its corresponding cross matrix. After computing the IF of the CIP field, we can back-propagate it for the purpose of maximizing or minimizing QMI.

$$
\begin{aligned}
&C^k = \{c_{ij}^k\}\ ,\ c_{ij}^k = v_{ij}^k - v_i^k - v_j^k + v^k\ ,\ k = 1, 2 \\
&v_c = \frac{1}{N^2}\sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}^k v_{ij}^l \\
&f_i^l = \frac{\partial v_c}{\partial a_i^l} = \frac{-1}{N^2\sigma^2}\sum_{j=1}^{N} c_{ij}^k v_{ij}^l d_{ij}^l \\
&i = 1, \dots, N,\quad l \neq k\ ,\quad l = 1, 2
\end{aligned}
\tag{7}
$$

4. Training MLPs with CIP

As Figure 2 shows, a MLP can be regarded as a communication channel. Each hidden layer in a MLP is one stage of the channel. The purpose of the training of a MLP is to transmit as much information as possible about the desired signal at each stage or layer. So, from this point of view, different layers can be trained one by one. First, the input layer can be trained so that the mutual information between the output of the layer and the desired signal is maximized. Then the hidden layers and the output layer can be trained in the same way respectively.

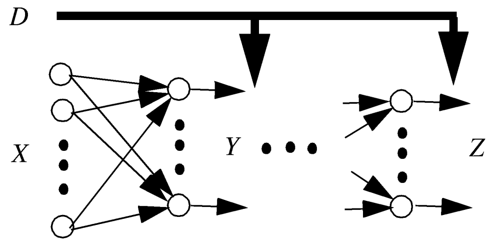

Figure 2. MLP as a communication channel.
X: input signal Y: hidden layer signal
Z: output signal D: desired signal

As an example, the input layer is $R^n \to R^m$: $Y = q(X, \theta)$, where θ is the weight set of the layer, and the training set is $\{(X_i, D_i)|i = 1, \dots N\}$, where X_i are input patterns and D_i are corresponding desired output signals, then there are actual outputs Y_i corresponding to each input pattern X_i. The mutual information between the output Y and the desired signal D can be estimated by cross information potential $V_c(b)$, where $b = \{(Y_i, D_i)|i = 1, \dots, N\}$ is the data set for CIP. To maximize the CIP, the gradient method can be used and the gradient can be calculated as:

$$
\frac{\partial}{\partial\theta}V_c(b) = \sum_{i=1}^{N}(f_i)^T\frac{\partial Y_i}{\partial\theta},\qquad f_i = \frac{\partial}{\partial Y_i}V_c(b)
\tag{8}
$$

where $\partial Y_i / \partial \theta$ are sensitivities, f_i are information forces in CIP field for each data point Y_i, and the calculation of them are exactly the same as described above.

To test the method, the problem of "frequency doubler" is tried, where the input signal is a sine wave and the desired output signal is still a sine wave but with its frequency doubled (as shown in Figure 3). A focused TDNN with one hidden layer is used. There are one input node with 5 delay taps in input, two nodes in hidden layer with tanh nonlinear function and one linear output node (as shown in Figure 4). The training scheme described above is used. The hidden layer is trained first followed by the output layer. The training curves are shown in Figure 5. The output of the hidden nodes and output node after training are shown in Figure 6 which tells us that the frequency of the final output is doubled.

5. Discussion and Conclusion

From the above experiment, we can see that even without the involvement of the output layer, CIP can still guide the hidden layer to learn what is needed. The plot of two hidden node outputs already reveals the doubled frequency which means the hidden nodes best represent the desired output. The output layer simply selects what is needed. These results, on the other hand, further confirm the validity of the CIP method proposed.

From the training curves, we can see the sharp increases in CIP which suggest that the step size should be varied and adapted during the training process. How to choose the kernel size of Gaussian function in CIP method is still an open problem. For these results, it is determined experimentally.

Acknowledgment: This work was partially supported by DARPA Grant F33615-97-1-1019.

References

[1] D. Xu, J. Principe, J. Fisher. H-C. Wu "A Novel Measure for Independent Component Analysis (ICA)" ICASSP'98, pp 1161-1164. 1998, Seattle.
[2] Dongxin Xu, Jose C. Principe "Learning from Examples with Quadratic Mutual Information" Proceedings of 1998 Workshop on Neural Networks for Signal Processing VIII, 1998, pp155-164.
[3] M.L.Minsky and S.A.Papert, 1969. Perceptrons. Cambridge, MA: MIT Press.
[4] Hecht-Nielsen,R., "Kolmogorov's mapping neural network existence theorem" 1st IEEE Int. Conf. on Neural Networks, vol 3, pp 11-14, 1987, San Diego.
[5] Zipser and Rumelhart. "Neurobiological Significance of new learning models", in E. Schwartz (Ed.), Computational Neuroscience, Cambridge, MA. MIT Press, 1990
[6] Linsker R. "An application of the principle of maximum information preservation to linear systems", in Advances in Neural Information Processing System 1, Morgan-Kaufman, pp 485-494, 1988
[7] Rumelhart, D.E., Hinton, G.E. and Williams, J.R. "Learning representations by back-propagating errors", Nature (London), 323, 1986, pp533-536.

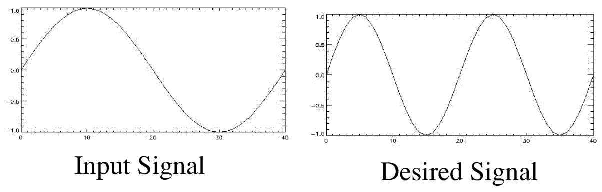

Figure 3. The Problem of "Frequency Doubler"

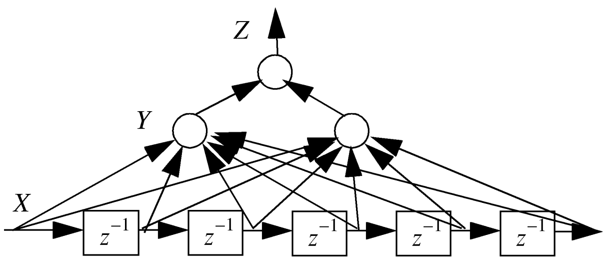

Figure 4. Focused TDNN as Frequency Doubler.

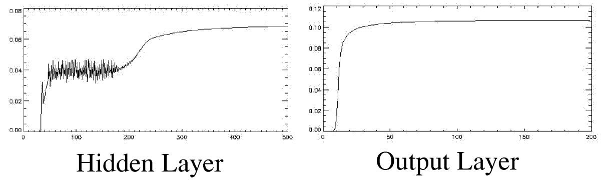

Figure 5. Training Curve. CIP vs. Iterations

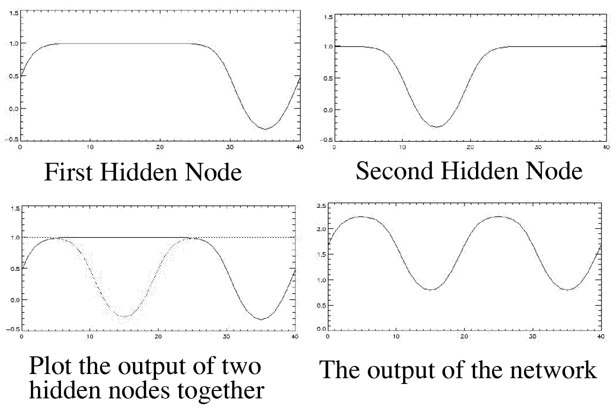

Figure 6. The output of the nodes after training